```
                        ;
                        ;       CP/M v1.4 cbios for TEI FDC-2 Controller board
                        ;
                        ;       CUTER ROM at C000
                        ;       Special disk rom at F000
                        ;
                        ;
0010 =          msize   equ     16              ;memory in k
F000 =          dskrom  equ     0f000h
3E00            bios    org     (msize*1024)-200h
3106 =          bdos    equ     bios-0d00h+6    ;bdos entry point
2900 =          ccp     equ     bios-1500h      ;command processor entry
002A =          nsects  equ     (bios-ccp)/128  ;length to warm boot
0003 =          iobyte  equ     3               ;special port byte
C019 =          sout    equ     0c019h          ;CUTER output
C01F =          sinp    equ     0c01fh          ;CUTER input
0014 =          comport equ     14h             ;comprint printer
0004 =          serport equ     4h              ;serial
                        ;
                        ;       standard bios transfer vector
                        ;
3E00 C3333E             jmp     boot            ;initial boot
3E03 C3583E     wboot   jmp     warm            ;warm boot
3E06 C3633F             jmp     const           ;console status
3E09 C3783F             jmp     conin           ;console read
3E0C C38E3F             jmp     conout          ;console write
3E0F C3A93F             jmp     list            ;list output
3E12 C3CA3F             jmp     punch           ;punch output
3E15 C3BE3F             jmp     reader          ;reader input
3E18 C3E13E             jmp     home            ;restore disk
3E1B C3CA3E             jmp     select          ;select disk
3E1E C3CF3E             jmp     settrk          ;set track number
3E21 C3D43E             jmp     setsec          ;set sector number
3E24 C3D93E             jmp     setdma          ;set buffer address
3E27 C3333F             jmp     read            ;read disk
3E2A C34B3F             jmp     write           ;write disk
3E2D C32D3E             jmp     $
3E30 C3303E             jmp     $
                        ;
                        ;       disk rom transfer vector
                        ;
F003 =          dskboot equ     dskrom+3        ;general boot
F006 =          dskinit equ     dskrom+6        ;init controller
F009 =          dsksel  equ     dskrom+9        ;pysical select
F00C =          dskrest equ     dskrom+12       ;restore to track 0
F00F =          dskseek equ     dskrom+15       ;physical seek
F012 =          dskrd   equ     dskrom+18       ;read sector
F015 =          dskwr   equ     dskrom+21       ;write sector
F018 =          dskrt   equ     dskrom+24       ;read track
F01B =          dskwt   equ     dskrom+27       ;write track (format)
F01E =          dskrtrk equ     dskrom+30       ;read track #
F021 =          dskwtrk equ     dskrom+33       ;set track #
                        ;
```

```
                ;       cold boot initialization
                ;
3E33 =          boot    equ     $
3E33 AF                 xra     a               ;init line printer
3E34 D314               out     comport         ;init pia
3E36 3D                 dcr     a
3E37 D315               out     comport+1
3E39 3E3E               mvi     a,3eh
3E3B D314               out     comport
3E3D DB15               in      comport+1       ;clear status
3E3F 3EFF               mvi     a,0ffh          ;send a null
3E41 D315               out     comport+1
3E43 3E36               mvi     a,36h           ;toggle strobe bit
3E45 D314               out     comport
3E47 3E3E               mvi     a,3eh
3E49 D314               out     comport
3E4B 3E81               mvi     a,10000001b     ;initialize iobyte
3E4D 320300             sta     iobyte
3E50 3E00               mvi     a,0             ;clear char buffers
3E52 32773F             sta     char
3E55 C39F3E             jmp     gocpm           ;static initialization
                ;
                ;       warm boot - read bdos and ccp
                ;
3E58 =          warm    equ     $
3E58 310001             lxi     sp,100h         ;reset stack
3E5B CD06F0             call    dskinit         ;reset controller
3E5E 0E00               mvi     c,0             ;drive 0 for boot
3E60 CDCA3E             call    select
3E63 CDE13E             call    home
3E66 C2583E             jnz     warm            ;no go...
3E69 062A               mvi     b,nsects        ;sectors to read
3E6B 0E00               mvi     c,0             ;current track
3E6D 1602               mvi     d,2             ;current sector
3E6F 210029             lxi     h,ccp           ;load address
3E72 =          warm1   equ     $
3E72 C5                 push    b               ;save registers
3E73 D5                 push    d
3E74 E5                 push    h
3E75 4A                 mov     c,d             ;get sector
3E76 CDD43E             call    setsec
3E79 C1                 pop     b               ;get dma address
3E7A C5                 push    b               ;save for later
3E7B CDD93E             call    setdma          ;set read address
3E7E CD333F             call    read            ;read it
3E81 C2583E             jnz     warm            ;error - try again
3E84 E1                 pop     h               ;restore dma address
3E85 118000             lxi     d,128           ;sector length
3E88 19                 dad     d               ;new dma address in hl
3E89 D1                 pop     d               ;get sector
3E8A C1                 pop     b               ;restore counter, track
3E8B 05                 dcr     b               ;1 less sector to go
3E8C CA9F3E             jz      gocpm           ;all set
3E8F 14                 inr     d               ;next sector
3E90 7A                 mov     a,d
```

```
3E91 FE1B               cpi     27              ;next track?
3E93 DA723E             jc      warm1           ;nope
3E96 1601               mvi     d,1             ;reset sector to 1
3E98 0C                 inr     c               ;increment track
3E99 CDCF3E             call    settrk          ;set track number
3E9C C3723E             jmp     warm1           ;and continue
                ;
                ;       transfer control to CP/M
                ;
3E9F =          gocpm   equ     $
3E9F 018000             lxi     b,80h           ;default buffer address
3EA2 CDD93E             call    setdma
3EA5 3EC3               mvi     a,0c3h          ;set up jumps
3EA7 320000             sta     0
3EAA 320500             sta     5
3EAD 21033E             lxi     h,wboot
3EB0 220100             shld    1
3EB3 210631             lxi     h,bdos
3EB6 220600             shld    6
3EB9 3EC3               mvi     a,0c3h          ;loop at rst 7
3EBB 323800             sta     070q
3EBE 213800             lxi     h,070q
3EC1 223900             shld    071q
3EC4 3E00               mvi     a,0             ;current drive to ccp
3EC6 4F                 mov     c,a
3EC7 C30029             jmp     ccp             ;away we go.....
                ;
                ;       set desired drive
                ;
3ECA =          select  equ     $
3ECA 79                 mov     a,c
3ECB 32E13F             sta     drvnxt
3ECE C9                 ret
                ;
                ;       set desired track
                ;
3ECF =          settrk  equ     $
3ECF 79                 mov     a,c
3ED0 32E73F             sta     track
3ED3 C9                 ret
                ;
                ;       set desired sector
                ;
3ED4 =          setsec  equ     $
3ED4 79                 mov     a,c
3ED5 32E83F             sta     sector
3ED8 C9                 ret
                ;
                ;       set desired buffer address
                ;
3ED9 =          setdma  equ     $
3ED9 E5                 push    h               ;save registers
3EDA 60                 mov     h,b
3EDB 69                 mov     l,c
3EDC 22E93F             shld    dma
```

```
3EDF E1                 pop     h               ;restore
3EE0 C9                 ret
                ;
                ;       send a disk (physically) home
                ;
3EE1 =          home    equ     $
3EE1 0E00               mvi     c,0             ;set track to 0
3EE3 CDCF3E             call    settrk
3EE6 CDED3E             call    seldsk          ;select the disk
3EE9 CD293F             call    seek            ;do the seek
3EEC C9                 ret
                ;
                ;       select disk for transfer
                ;
3EED =          seldsk  equ     $
3EED C5                 push    b               ;save registers
3EEE 3AE03F             lda     drvcur          ;what was last drive?
3EF1 4F                 mov     c,a
3EF2 3AE13F             lda     drvnxt          ;what is desired drive?
3EF5 B9                 cmp     c               ;save drive?
3EF6 CA213F             jz      seldsk2         ;yes
3EF9 E5                 push    h               ;save register
3EFA 21E23F             lxi     h,drvtrk        ;point to track table
3EFD 0600               mvi     b,0
3EFF 09                 dad     b               ;index by current drive
3F00 CD1EF0             call    dskrtrk         ;get head position
3F03 77                 mov     m,a             ;save current position
3F04 3AE13F             lda     drvnxt          ;get desired drive
3F07 4F                 mov     c,a
3F08 CD09F0             call    dsksel          ;select it for real
3F0B 21E23F             lxi     h,drvtrk        ;point to track table
3F0E 09                 dad     b               ;index by drive
3F0F 7E                 mov     a,m             ;retrieve its position
3F10 4E                 mov     c,m
3F11 FEFF               cpi     0ffh            ;is it known??
3F13 C21C3F             jnz     seldsk1         ;yes it is
3F16 CD0CF0             call    dskrest         ;no - calibrate
3F19 0E00               mvi     c,0             ;set track to 0
3F1B 71                 mov     m,c
3F1C =          seldsk1 equ     $
3F1C 79                 mov     a,c             ;get position
3F1D CD21F0             call    dskwtrk         ;tell the controller
3F20 E1                 pop     h               ;restore register
3F21 =          seldsk2 equ     $
3F21 3AE13F             lda     drvnxt          ;set current drive
3F24 32E03F             sta     drvcur
3F27 C1                 pop     b               ;restore registers
3F28 C9                 ret
                ;
                ;       do a seek
                ;
3F29 =          seek    equ     $
3F29 C5                 push    b               ;save register
3F2A 3AE73F             lda     track           ;get desired track
3F2D 4F                 mov     c,a
```

```
3F2E CD0FF0              call    dskseek         ;do the seek
3F31 C1                  pop     b               ;restore register
3F32 C9                  ret
                 ;
                 ;       do a read
                 ;
3F33 =           read    equ     $
3F33 E5                  push    h               ;save registers
3F34 C5                  push    b
3F35 CDED3E              call    seldsk          ;select disk
3F38 CD293F              call    seek            ;position head, load
3F3B C2483F              jnz     read1           ;error
3F3E 2AE93F              lhld    dma             ;get buffer address
3F41 3AE83F              lda     sector          ;get desired sector
3F44 4F                  mov     c,a
3F45 CD12F0              call    dskrd           ;do the read
3F48 =           rea`1   equ
3F48 C1                  pop     b               ;restore registers
3F49 E1                  pop     h
3F4A C9                  ret
                 ;
                 ;       do a write
                 ;
3F4B =           write   equ     $
3F4B E5                  push    h               ;save registers
3F4C C5                  push    b
3F4D CDED3E              call    seldsk          ;select drive
3F50 CD293F              call    seek            ;position head, load
3F53 C2603F              jnz     write1          ;error
3F56 2AE93F              lhld    dma             ;get buffer address
3F59 3AE83F              lda     sector          ;get desired sector
3F5C 4F                  mov     c,a
3F5D CD15F0              call    dskwr           ;write it out
3F60 =           write1  equ     $
3F60 C1                  pop     b               ;restore registers
3F61 E1                  pop     h
3F62 C9                  ret
                 ;
                 ;       keyboard status
                 ;
                 const:
3F63 3A773F              lda     char            ;see if char present
3F66 B7                  ora     a
3F67 3E00                mvi     a,0             ;set up for cma
3F69 C2753F              jnz     kbdst1          ;already have a char
3F6C CD1FC0              call    sinp            ;call status/get
3F6F 32773F              sta     char            ;save char if any
3F72 3E00                mvi     a,0             ;clear status
3F74 C8                  rz
                 kbdst1:
3F75 2F                  cma                     ;no char available
3F76 C9                  ret
3F77 00          char:   db      0               ;character buffer
                 ;
                 ;       keyboard input
```

```
                ;
                conin:
3F78 3A773F             lda     char            ;get char
3F7B B7                 ora     a               ;test if present
3F7C C2853F             jnz     kbdin1          ;have char already
3F7F CD633F             call    const           ;test for char
3F82 C3783F             jmp     conin           ;and try again
                kbdin1:
3F85 4F                 mov     c,a             ;save char
3F86 AF                 xra     a               ;clear buffer
3F87 32773F             sta     char
3F8A 79                 mov     a,c             ;restore char
3F8B E67F               ani     7fh             ;strip parity
3F8D C9                 ret
                ;
                ;       video output
                ;
                conout:
3F8E C5                 push    b               ;save registers
3F8F 41                 mov     b,c             ;current char in b
3F90 3AA33F             lda     lastchar        ;get last char
3F93 CDA43F             call    checkcr         ;ignore duplicate cr
3F96 CAA03F             jz      noprint
3F99 79                 mov     a,c             ;get char in a
3F9A 32A33F             sta     lastchar        ;save as last
3F9D CD19C0             call    sout            ;display char
                noprint:
3FA0 C1                 pop     b               ;restore registers
3FA1 79                 mov     a,c             ;set a to char
3FA2 C9                 ret                     ;and return
3FA3 00         lastchar db     0               ;last char displayed
                ;
                ;       routine to check for two cr in a row
                ;
                checkcr:
3FA4 FE0D               cpi     0dh             ;was last char a cr
3FA6 C0                 rnz                     ;no, print this one
3FA7 B8                 cmp     b               ;is this one a cr too?
3FA8 C9                 ret                     ;return with status
                ;
                ;       printer output
                ;
                list:
3FA9 DB14               in      comport         ;get status
3FAB 07                 rlc                     ;test status
3FAC D2A93F             jnc     list            ;wait
3FAF DB15               in      comport+1       ;clear status
3FB1 79                 mov     a,c             ;get char
3FB2 2F                 cma                     ;invert for comprint
3FB3 D315               out     comport+1
3FB5 3E36               mvi     a,36h           ;toggle strobe bit
3FB7 D314               out     comport
3FB9 3E3E               mvi     a,3eh
3FBB D314               out     comport
3FBD C9                 ret
```

```
                ;
                ;       serial input
                ;
                reader:
3FBE DB04               in      serport         ;get status
3FC0 E601               ani     1               ;test status
3FC2 C2BE3F             jnz     reader          ;wait
3FC5 DB05               in      serport+1       ;get char
3FC7 E67F               ani     7fh             ;clear parity
3FC9 C9                 ret
                ;
                ;       serial output
                ;
                punch:
3FCA DB04               in      serport         ;get status
3FCC E680               ani     80h             ;test status
3FCE C2CA3F             jnz     punch           ;wait
3FD1 79                 mov     a,c             ;get char
3FD2 D304               out     serport         ;send it
3FD4 C9                 ret
                ;
                ;       scratch area
                ;
3FE0            scratch org     msize*1024-32   ;top of memory
3FE0 05         drvcur  db      5               ;max drive + 1
3FE1 00         drvnxt  db      0
3FE2 FFFFFFFFFFdrvtrk   db      0ffh,0ffh,0ffh,0ffh,0ffh  ;track table
3FE7 00         track   db      0               ;desired track
3FE8 00         sector  db      0               ;desired sector
3FE9 8000       dma     dw      80h             ;buffer address
3FEB                    end
```